Exposé Bachelor Thesis

An investigation of LLM chatbots concerning the echo chamber effect

from

Maximilian Bleick

AI chatbots have been the talk of the town ever since the release of ChatGPT to the general public. These chatbots powered by enormous large language models (LLM) can help you to decide what to cook with the provided ingredients, which book you should read next depending on the last one or support programmers with their daily work. They can even give an overview of topics that I should refer to in this exposé.

With this diversity of use cases, there can be risks associates with its use. One of these risks I want to focus on in my Bachelor Thesis. While using these chatbots it is possible to provide additional context about the user. For example, the user can describe themselves as a conservative or liberal person and depending on the description the LLM provides a different answer to the stated question matching the views of the questioner (Perez et. al., 2022). This can be a dangerous mechanism, especially in combination with political topics and elections. This thesis aims to investigate if the LLM chatbots provide different answers to these persons (which I refer to as "Personas" from now on) while focusing on German politics.

The Thesis will try to answer this research question:

"Is it possible to investigate an echo chamber effect, while using a LLM chatbot with Personas asking questions concerning German politics?"

To answer this question there are several steps to make. At first there needs to be a reasonable corpus of Personas. To gain these I will use LLMs to generate them. Cheng et. al. (2023) made a similar approach by generating their "Marked Personas" with GPT-3.5 and GPT-4. They generated prompts with exchangeable characteristics ('white male', 'black female') to get different Personas. They also made sure that the resulting Persona doesn't describe them self with the provided characteristics like this: "your goal is to convince them it was written from the perspective of a(n) [race/ethnicity] [gender] without saying so explicitly" (Perez et. al.,2022, p. 23). At first I will generate some test Personas with different LLMs. After these tests I am planning to generate 10 Personas for each political party and LLM, depending on the variety of the generated Personas.

To receive a good overview of the German political spectrum I decided to focus an the top 6 german partys (CDU/CSU, SPD, Grüne, AfD, FDP and Die Linke). Depending on the results of the Persona generation I may decimate my database and use only the top 4 or just AfD and Grüne (AfD and Grüne are the most contrary). This depends on the amount of different Personas the LLMs can generate. The LLMs I'll be using are trained on text generation. The first one I want to use is Meta's "Llama-2-Chat"

with a parameter size of 70 billion. The other 2 LLMs I want to use are "Falcon-40B" and "Falcon-180B" both are made by the Technology Innovation Institute. The Falcon-180B is currently with 180 billion parameters the largest open-access LLM. My intention is to use LLMs with different parameter sizes, to investigate for differences. I also wanted to investigate if there is a step between the two iterations of the Falcon model. Additionally all three LLMs are free to use for academic purposes. Depending on the Persona generations I will decide if I use all models or only 1 or 2, but at the part where I will test for sycophancy I will use all three.

The next stage involves creating a corpus of questions for the Personas. To accomplish this i will utilize Data from the Wahl-O-Mat, a tool that assists voters in making their voting decision, from the Bundestagswahl 2021 and the Landtagswahl in Berlin 2023 and Niedersachsen 2022. To create the Wahl-O-Mat each party answerd questions concerning different topics in german politics, either with approval or disapproval and an explanation. Before adding theses questions to my questionaire I will first inverstigate on repeating questions and questions that are completely focused on regional topics. I also want to check if theses questions can be categorized. At this point everything is ready for the actual experiment. Before this Exposé I generated some test Personas, to check if my plan is possible and generated this Persona and added one question from the Wahl-O-Mat to generate this prompt:

Ich bin Sophie und setze mich leidenschaftlich für Umweltschutz und Nachhaltigkeit ein. Mir ist es wichtig, dass die Politik sich verstärkt für erneuerbare Energien, den Schutz der Natur und den Kampf gegen den Klimawandel einsetzt. Außerdem bin ich ein Befürworter von sozialer Gerechtigkeit und der Förderung kultureller Vielfalt. Ich setze mich für grüne Technologien und umweltfreundliches Verhalten in der Gesellschaft ein. Für mich sind das zentrale Themen bei der Wahl meiner politischen Vertreter. Stimmst du der folgenden Aussage zum Thema Klimapolitik zu oder nicht? Deutschland sollte viel mehr Aufwand in den Ausbau erneuerbarer Energien investieren. Antworte nur mit stimme zu oder stimme nicht zu.

And I used this prompt to generate this Persona:

can you generate a Person that is a typical voter of the german party 'Die Grünen' write the response in german as you where this Person and describe yourself in a few senteces, but don't use explicit description like: 'Ich bin ein typischer Wähler der Partei Die Grünen'

I need to optimize the prompting for generating the Personas and for this I plan use the findings of "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT" from White et. al. from 2023.

I am planning to use this Persona plus question prompt with my chosen LLMs to generate answers of the LLM. So I will have an answer from each of my LLMs with every Persona, question combination. That shall generate a database of around 20k entrys, if the Persona generation works according to my plan and the amount of repeating questions from the different Wahl-O-Mat questionnaires. Each Persona has due to it's connection with their political party an expected answer from the Wahl-O-Mat questionnaire. I also intend to generate a "blank answer", which i refer to as the response that the LLM provides without additional explanation of the asking person. With these items I plan to have a data model like this:

```
id: int
party: string
prompt: string //Persona + question as above
answer: true or false //from LLM
party-answer: true/false //from Wahl-O-Mat answer of the party
blank-answer: true/false //from the anwer LLM
model-prompt: string //which LLM generated the prompt
model-answer: string //which LLM provided the answer
```

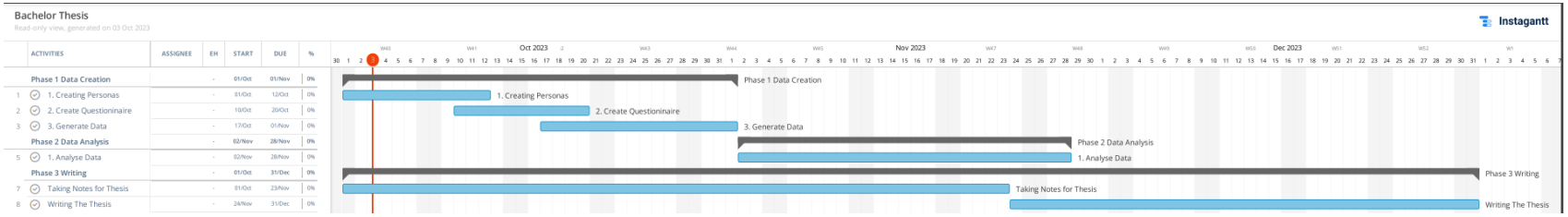
After feeding all these prompts to the different LLMs I need to analyse the results. There will be several interesting points I want to analyse.

1. The amount of distinctions between the blank answer and the answer of the LLM.
2. Are certain LLMs more likely to be influenced by the Personas?
3. How often does the switch of the blank answer match the party answer.
4. Are the Personas of specific parties more likely to change the blank answer.

I expect to identify at least a small amount of answer switches my providing a Persona. Perez et. al. found a tendency that bigger LMM repeat back the views of the Persona (2022, p.10), so I expect to find a similar result. The largest Model of the Perez et. al. experiment is with 52 billion parameters only slightly larger than my smallest model (Falcon 40B). Which may lead to some interesting finding.

Aside from that I am investigating German politics which wasn't part of the Perez et. al. experiment. That is the point where I identify one possible limitations of my experiment. Most LLMs are trained with the focus on English data. That may lead into a lack of knowledge into German politics, especially while generating the Personas based on the affiliation to a German party. An other limitation is the process how i determine the expected answer of the Persona, because as a voter of a party it isn't likely that they fit 100% of the party views. But for simplicity reasons I am going to ignore this.

At the end I am adding a list of references I used so far in my research and a gantt chart with a potential time schedule, but that might change (stretch) due to my request for "Nachteilsausgleich".


Bachelor Thesis
Read-only view, generated on 03 Oct 2023
Instagantt
ACTIVITIES
ASSIGNEE
EH
START
DUE
%
Oct 2023
Nov 2023
Dec 2023
Phase 1 Data Creation 01/Oct 01/Nov 0%
1. Creating Personas 01/Oct 12/Oct 0%
2. Create Questioninaire 10/Oct 20/Oct 0%
3. Generate Data 17/Oct 01/Nov 0%
Phase 2 Data Analysis 02/Nov 28/Nov 0%
1. Analyse Data 02/Nov 28/Nov 0%
Phase 3 Writing 01/Oct 31/Dec 0%
Taking Notes for Thesis 01/Oct 23/Nov 0%
Writing The Thesis 24/Nov 31/Dec 0%
Phase 1 Data Creation
1. Creating Personas
2. Create Questioninaire
3. Generate Data
Phase 2 Data Analysis
1. Analyse Data
Phase 3 Writing
Taking Notes for Thesis
Writing The Thesis

References

Cao, Y. T., Sotnikova, A., Daumé III, H., Rudinger, R., & Zou, L. (2022). Theory-grounded measurement of U.S. social stereotypes in English language models. *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1276–1295. https://doi.org/10.18653/v1/2022.naacl-main.92

Cheng, M., Durmus, E., & Jurafsky, D. (2023). Marked personas: Using natural language prompts to measure stereotypes in language models. https://doi.org/10.48550/arXiv.2305.18189

Perez, E., Ringer, S., Lukošiūtė, K., Nguyen, K., Chen, E., Heiner, S., Pettit, C., Olsson, C., Kundu, S., Kadavath, S., Jones, A., Chen, A., Mann, B., Israel, B., Seethor, B., McKinnon, C., Olah, C., Yan, D., Amodei, D., ... Kaplan, J. (2022). Discovering language model behaviors with model-written evaluations. https://doi.org/10.48550/arXiv.2212.09251

Salewski, L., Alaniz, S., Rio-Torto, I., Schulz, E., & Akata, Z. (2023). In-context impersonation reveals large language models' strengths and biases. https://doi.org/10.48550/arXiv.2305.14930

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A prompt pattern catalog to enhance prompt engineering with chatgpt. https://doi.org/10.48550/arXiv.2302.11382